<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>OnWard Global-Trade-Specialists Survey</title>

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">

    <script src="https://cdn.tailwindcss.com"></script>

    <style>

        :root {

            --onward-dark-blue: #0A2342;

            --onward-accent: #00BFFF;

        }

        body {

            font-family: 'Inter', sans-serif;

            color: #334155;

        }

        .bg-onward-blue {

            background-color: var(--onward-dark-blue);

        }

        .text-onward-blue {

            color: var(--onward-dark-blue);

        }

        .btn-onward {

            background-color: var(--onward-dark-blue);

            color: white;

            transition: background-color 0.3s, transform 0.2s;

        }

        .btn-onward:hover {

            background-color: #1a659e; /\* Slightly lighter shade for hover \*/

            transform: translateY(-2px);

        }

        .btn-onward:disabled {

            background-color: #94a3b8;

            cursor: not-allowed;

        }

        .border-onward {

            border-color: var(--onward-dark-blue);

        }

        /\* Styles for the background slideshow \*/

        .background-slideshow {

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            z-index: -1;

            background-color: var(--onward-dark-blue);

        }

        .background-slideshow img {

            position: absolute;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            object-fit: cover;

            opacity: 0;

            transition: opacity 1.5s ease-in-out;

        }

        .background-slideshow img.active {

            opacity: 1;

        }

        .back-btn {

            background-color: #e2e8f0;

            color: #475569;

            transition: background-color 0.3s;

        }

        .back-btn:hover {

            background-color: #cbd5e1;

        }

        .data-card {

            border-radius: 0.5rem;

            padding: 1rem;

            margin-bottom: 1rem;

            background-color: #f8fafc;

        }

        /\* New styles for centered header \*/

        .onward-gts-header {

            display: flex;

            flex-direction: column;

            align-items: center;

        }

        .onward-gts-header h1 {

            margin: 0;

            line-height: 1;

        }

        .onward-gts-header p {

            margin: 0;

        }

        /\* New style for transparent pink card background \*/

        .bg-salmon-transparent {

            background-color: rgba(255, 192, 203, 0.7);

        }

    </style>

</head>

<body class="flex items-center justify-center min-h-screen p-4">

    <!-- Background Slideshow -->

    <div class="background-slideshow">

        <img class="active" src="wetransfer\_onward-pull-up-banners\_2025-08-18\_0917.zip/Onward Poster/1.png" alt="Onward Poster 1">

        <img src="wetransfer\_onward-pull-up-banners\_2025-08-18\_0917.zip/Onward Poster/2.png" alt="Onward Poster 2">

        <img src="https://placehold.co/1920x1080/0A2342/ffffff?text=Truck+on+the+Road" alt="Truck on the Road">

        <img src="https://placehold.co/1920x1080/0A2342/ffffff?text=Cargo+Plane+Flying" alt="Cargo Plane Flying">

    </div>

    <div class="bg-salmon-transparent p-8 md:p-12 rounded-xl shadow-lg w-full max-w-2xl backdrop-blur-sm relative">

        <!-- Admin Login Link -->

        <button id="admin-link" class="absolute top-4 right-4 text-sm text-gray-500 hover:text-onward-blue">Admin Login</button>

        <!-- Header Section -->

        <div class="text-center mb-8 onward-gts-header">

            <h1 class="text-3xl md:text-4xl font-bold text-onward-blue">OnWard</h1>

            <p class="mt-0 text-gray-500 text-lg md:text-xl font-semibold">Global-Trade-Specialists</p>

        </div>

        <!-- App Container -->

        <div id="app-container">

            <!-- Content will be dynamically injected here -->

        </div>

        <!-- Admin Dashboard Container -->

        <div id="admin-dashboard" class="hidden">

            <h2 class="text-2xl font-bold text-onward-blue mb-4">Admin Dashboard</h2>

            <div id="submissions-list" class="space-y-4">

                <p class="text-center text-gray-500">Loading submissions...</p>

            </div>

            <div class="flex justify-center mt-6">

                <button id="logout-btn" class="back-btn py-2 px-6 rounded-lg shadow-md">Logout</button>

            </div>

        </div>

        <!-- Admin Login Form -->

        <div id="admin-login-form" class="hidden">

            <div class="flex flex-col items-center">

                <h2 class="text-2xl font-bold text-onward-blue mb-4">Admin Login</h2>

                <form id="login-form-fields" class="w-full max-w-sm">

                    <div class="mb-4">

                        <label for="username" class="block text-gray-700 font-bold mb-2">Username:</label>

                        <input type="text" id="username" name="username" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring-2 focus:ring-onward-blue">

                    </div>

                    <div class="mb-6">

                        <label for="password" class="block text-gray-700 font-bold mb-2">Password:</label>

                        <input type="password" id="password" name="password" class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring-2 focus:ring-onward-blue">

                    </div>

                    <div class="flex items-center justify-between">

                        <button type="submit" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Login</button>

                    </div>

                    <p id="login-error" class="text-red-500 text-sm mt-4 hidden"></p>

                </form>

                <button id="back-to-survey-btn" class="back-btn mt-6 py-2 px-6 rounded-lg shadow-md">Back to Survey</button>

            </div>

        </div>

        <!-- Confirmation Message -->

        <div id="confirmation" class="hidden text-center">

            <h2 class="text-2xl font-bold text-onward-blue mb-4">Thank You!</h2>

            <p class="text-gray-600 mt-4">Your feedback has been submitted and saved securely. We appreciate your time and input.</p>

            <div class="flex justify-center mt-6">

                <button id="start-new-survey" class="btn-onward py-2 px-6 rounded-lg shadow-md">Start New Survey</button>

            </div>

            <div id="user-id-display" class="mt-4 text-sm text-gray-500"></div>

        </div>

    </div>

    <script type="module">

        import { initializeApp } from "https://www.gstatic.com/firebasejs/10.11.0/firebase-app.js";

        import { getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/10.11.0/firebase-auth.js";

        import { getFirestore, addDoc, collection, setLogLevel, getDocs } from "https://www.gstatic.com/firebasejs/10.11.0/firebase-firestore.js";

        document.addEventListener('DOMContentLoaded', () => {

            setLogLevel('debug');

            const appContainer = document.getElementById('app-container');

            const adminLoginView = document.getElementById('admin-login-form');

            const adminDashboard = document.getElementById('admin-dashboard');

            const confirmation = document.getElementById('confirmation');

            const slideshowImages = document.querySelectorAll('.background-slideshow img');

            const submissionsList = document.getElementById('submissions-list');

            const userIdDisplay = document.getElementById('user-id-display');

            const adminLink = document.getElementById('admin-link');

            const backToSurveyBtn = document.getElementById('back-to-survey-btn');

            const startNewSurveyBtn = document.getElementById('start-new-survey');

            const logoutBtn = document.getElementById('logout-btn');

            let currentIndex = 0;

            let db;

            let auth;

            let userId;

            const ADMIN\_USERNAME = "admin";

            const ADMIN\_PASSWORD = "onwardgts";

            const appId = typeof \_\_app\_id !== 'undefined' ? \_\_app\_id : 'default-app-id';

            const firebaseConfig = typeof \_\_firebase\_config !== 'undefined' ? JSON.parse(\_\_firebase\_config) : {};

            const initialAuthToken = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

            let surveyState = {

                currentStep: 0,

                role: null,

                responses: {}

            };

            const surveyQuestions = {

                'Seller': {

                    'part1': {

                        'title': 'What type of products do you sell?',

                        'field': 'productType',

                        'options': ['Handmade / Manufactured', 'Resale', 'Digital products'],

                        'type': 'checkbox',

                        'nextText': 'Where do you currently sell?',

                        'nextField': 'sellingChannel',

                        'nextOptions': ['Local only', 'Online (domestic)', 'Online (international)', 'In-person markets/events'],

                    },

                    'part2': {

                        'title': 'Export experience:',

                        'field': 'exportExperience',

                        'options': ['None', 'Some', 'Regular exports'],

                        'type': 'radio',

                        'nextText': 'Biggest challenges when selling to other countries (select all):',

                        'nextField': 'challenges',

                        'nextOptions': ['Finding buyers', 'Shipping costs', 'Customs paperwork', 'Payment solutions', 'Marketing and promotion'],

                        'nextNextText': 'Which services would you be interested in? (select all)',

                        'nextNextField': 'servicesInterested',

                        'nextNextOptions': ['Marketplace listing', 'Product photography', 'Translation services', 'Payment solutions', 'Bulk shipping discounts'],

                    }

                },

                'Transporter': {

                    'part1': {

                        'title': 'What type(s) of transport do you offer?',

                        'field': 'transportType',

                        'options': ['Road', 'Rail', 'Air', 'Sea'],

                        'type': 'checkbox',

                        'nextText': 'Carrying capacity (select all):',

                        'nextField': 'carryingCapacity',

                        'nextOptions': ['Small packages', 'Pallets', 'Containers', 'Bulk freight'],

                    },

                    'part2': {

                        'title': 'Current routes (select all):',

                        'field': 'currentRoutes',

                        'options': ['Local only', 'Cross-border (regional)', 'Global'],

                        'type': 'checkbox',

                        'nextText': 'Biggest challenges (select all):',

                        'nextField': 'challenges',

                        'nextOptions': ['Customs delays', 'Route optimization', 'Fuel costs', 'Finding clients', 'Tracking technology'],

                        'nextNextText': 'Which services would you be interested in? (select all):',

                        'nextNextField': 'servicesInterested',

                        'nextNextOptions': ['Client matching', 'Route optimization software', 'Bulk fuel discounts', 'Customs clearance', 'Tracking integration'],

                    }

                },

                'Warehouse Manager': {

                    'part1': {

                        'title': 'Type of facility (select all):',

                        'field': 'facilityType',

                        'options': ['Storage', 'Manufacturing', 'Both'],

                        'type': 'checkbox',

                        'nextText': 'Industry focus (select all):',

                        'nextField': 'industryFocus',

                        'nextOptions': ['Food', 'Textiles', 'Electronics', 'Machinery'],

                    },

                    'part2': {

                        'title': 'Capacity:',

                        'field': 'capacity',

                        'options': ['Small', 'Medium', 'Large'],

                        'type': 'radio',

                        'nextText': 'Automation level:',

                        'nextField': 'automationLevel',

                        'nextOptions': ['Manual', 'Semi-automated', 'Fully automated'],

                        'nextNextText': 'Services interested in (select all):',

                        'nextNextField': 'servicesInterested',

                        'nextNextOptions': ['Client matching', 'Automation upgrades', 'Export packaging', 'Compliance support', 'Global marketing'],

                    }

                },

                'Freight Forwarder': {

                    'part1': {

                        'title': 'Which services do you offer? (select all):',

                        'field': 'servicesOffered',

                        'options': ['Air freight', 'Sea freight', 'Road freight', 'Rail freight', 'Customs clearance', 'Warehousing', 'Documentation handling'],

                        'type': 'checkbox',

                    },

                    'part2': {

                        'title': 'Regions served (select all):',

                        'field': 'regionsServed',

                        'options': ['Domestic only', 'Regional', 'Continental', 'Global'],

                        'type': 'checkbox',

                        'nextText': 'Average shipment size:',

                        'nextField': 'averageShipmentSize',

                        'nextOptions': ['Small packages', 'Pallets', 'Containers', 'Bulk freight'],

                        'nextNextText': 'Challenges (select all):',

                        'nextNextField': 'challenges',

                        'nextNextOptions': ['Customs delays', 'Carrier capacity', 'Documentation errors', 'Finding clients'],

                        'nextNextNextText': 'Services interested in (select all):',

                        'nextNextNextField': 'servicesInterested',

                        'nextNextNextOptions': ['Client acquisition', 'Tracking integration', 'Trade compliance', 'Digital documentation'],

                    }

                },

                'Importer / Exporter': {

                    'part1': {

                        'title': 'Business type:',

                        'field': 'businessType',

                        'options': ['Importer', 'Exporter', 'Both'],

                        'type': 'radio',

                        'nextText': 'Type of goods (select all):',

                        'nextField': 'goodsType',

                        'nextOptions': ['Raw materials', 'Agricultural', 'Manufactured', 'Machinery & equipment', 'Consumer products'],

                    },

                    'part2': {

                        'title': 'Where do you currently trade? (select continents):',

                        'field': 'tradingContinents',

                        'options': ['Africa', 'Asia', 'Europe', 'North America', 'South America', 'Oceania'],

                        'type': 'checkbox',

                        'nextText': 'Challenges (select all):',

                        'nextField': 'challenges',

                        'nextOptions': ['High logistics costs', 'Finding reliable partners', 'Customs clearance', 'Payment collection'],

                        'nextNextText': 'Helpful services (select all):',

                        'nextNextField': 'servicesInterested',

                        'nextNextOptions': ['Market research', 'Shipping booking', 'Customs compliance support', 'Trade finance', 'Warehousing & distribution', 'Insurance'],

                    }

                },

                'Distributor': {

                    'part1': {

                        'title': 'Type of distribution (select all):',

                        'field': 'distributionType',

                        'options': ['Wholesale (B2B)', 'Retail (B2C)', 'Both'],

                        'type': 'checkbox',

                        'nextText': 'Industries served (select all):',

                        'nextField': 'industriesServed',

                        'nextOptions': ['Food & Beverage', 'Textiles & Apparel', 'Electronics', 'Machinery', 'Pharmaceuticals & Healthcare'],

                    },

                    'part2': {

                        'title': 'Network size:',

                        'field': 'networkSize',

                        'options': ['Local', 'Regional', 'Continental', 'Global'],

                        'type': 'radio',

                        'nextText': 'Challenges (select all):',

                        'nextField': 'challenges',

                        'nextOptions': ['Inventory management', 'Delivery/logistics costs', 'Finding reliable suppliers', 'Payment collection'],

                        'nextNextText': 'Services interested in (select all):',

                        'nextNextField': 'servicesInterested',

                        'nextNextOptions': ['Inventory tools', 'Logistics partnerships', 'Trade finance', 'Distribution integrations'],

                    }

                },

                'payment': {

                    'title': 'Preferred payment model (select one):',

                    'field': 'paymentModel',

                    'options': ['Monthly subscription', 'Annual', 'Freemium', 'Integrated payment processing (in-app payments)'],

                    'type': 'radio',

                    'nextText': 'If subscription/annual/freemium: How much would you pay per month?',

                    'nextField': 'monthlyFee',

                    'nextOptions': ['R0–R500', 'R500–R1 000', 'R1 000–R2 500', 'R2 500–R5 000', 'R5 000+'],

                }

            };

            // Initialize Firebase

            const app = initializeApp(firebaseConfig);

            db = getFirestore(app);

            auth = getAuth(app);

            // Function to handle the slideshow

            function startSlideshow() {

                setInterval(() => {

                    slideshowImages[currentIndex].classList.remove('active');

                    currentIndex = (currentIndex + 1) % slideshowImages.length;

                    slideshowImages[currentIndex].classList.add('active');

                }, 5000);

            }

            // Authentication state listener

            onAuthStateChanged(auth, async (user) => {

                if (user) {

                    userId = user.uid;

                    userIdDisplay.textContent = `Your user ID for secure storage: ${userId}`;

                    console.log('Authenticated with user ID:', userId);

                } else {

                    console.log('No user signed in. Attempting anonymous sign-in.');

                    await signInAnonymously(auth);

                }

            });

            // Initial authentication check

            async function authenticate() {

                try {

                    if (initialAuthToken) {

                        await signInWithCustomToken(auth, initialAuthToken);

                        console.log("Signed in with custom token.");

                    } else {

                        await signInAnonymously(auth);

                        console.log("Signed in anonymously.");

                    }

                } catch (error) {

                    console.error("Authentication error:", error);

                }

            }

            function goBack() {

                if (surveyState.currentStep > 0) {

                    surveyState.currentStep--;

                    renderStep();

                }

            }

            function showView(viewId) {

                appContainer.classList.add('hidden');

                adminLoginView.classList.add('hidden');

                adminDashboard.classList.add('hidden');

                confirmation.classList.add('hidden');

                document.getElementById(viewId).classList.remove('hidden');

            }

            function handleAdminLogin(event) {

                event.preventDefault();

                const username = document.getElementById('username').value;

                const password = document.getElementById('password').value;

                const errorEl = document.getElementById('login-error');

                if (username === ADMIN\_USERNAME && password === ADMIN\_PASSWORD) {

                    renderDashboard();

                } else {

                    errorEl.textContent = 'Invalid username or password.';

                    errorEl.classList.remove('hidden');

                }

            }

            async function renderDashboard() {

                showView('admin-dashboard');

                submissionsList.innerHTML = '<p class="text-center text-gray-500">Loading submissions...</p>';

                try {

                    // FIX: Changed the path to read from the current user's private collection

                    // to fix the "Missing or insufficient permissions" error.

                    // This will only show the current user's submissions.

                    const submissionsRef = collection(db, `artifacts/${appId}/users/${userId}/surveyResponses`);

                    const querySnapshot = await getDocs(submissionsRef);

                    const submissions = [];

                    querySnapshot.forEach(doc => {

                        submissions.push({ id: doc.id, ...doc.data() });

                    });

                    if (submissions.length > 0) {

                        submissionsList.innerHTML = submissions.map(sub => `

                            <div class="data-card shadow-lg">

                                <h3 class="font-bold text-onward-blue mb-2">Submission ID: ${sub.id}</h3>

                                <p class="text-xs text-gray-400 mb-2">${sub.timestamp?.toDate()?.toLocaleString() || 'N/A'}</p>

                                <div class="space-y-1">

                                    ${Object.keys(sub).filter(key => key !== 'timestamp' && key !== 'id').map(key => `

                                        <div class="flex">

                                            <span class="font-semibold text-gray-700 w-1/3">${key}:</span>

                                            <span class="text-gray-600 w-2/3">${Array.isArray(sub[key]) ? sub[key].join(', ') : sub[key]}</span>

                                        </div>

                                    `).join('')}

                                </div>

                            </div>

                        `).join('');

                    } else {

                        submissionsList.innerHTML = '<p class="text-center text-gray-500">No submissions found yet.</p>';

                    }

                } catch (error) {

                    console.error("Error fetching documents:", error);

                    submissionsList.innerHTML = '<p class="text-center text-red-500">Error loading data. Please try again.</p>';

                }

            }

            function renderStep() {

                showView('app-container');

                let html = '';

                const backButton = surveyState.currentStep > 0 ? `

                    <button id="back-btn" class="back-btn py-3 px-6 rounded-lg shadow-md mt-6">Back</button>

                ` : '';

                const state = surveyState;

                if (state.currentStep === 0) {

                    html = `

                        <div class="survey-step" id="step-0">

                            <p class="text-lg font-semibold mb-4">What is your primary role in the logistics industry?</p>

                            <div class="flex flex-col space-y-4">

                                <button data-role="Seller" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Seller</button>

                                <button data-role="Distributor" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Distributor</button>

                                <button data-role="Transporter" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Transporter</button>

                                <button data-role="Warehouse Manager" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Warehouse Manager</button>

                                <button data-role="Freight Forwarder" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Freight Forwarder</button>

                                <button data-role="Importer / Exporter" class="btn-onward w-full py-3 rounded-lg shadow-md focus:outline-none focus:ring-2 focus:ring-onward-blue focus:ring-offset-2">Importer / Exporter</button>

                            </div>

                        </div>

                    `;

                } else if (state.currentStep === 1) {

                    const data = surveyQuestions[state.role].part1;

                    html = `

                        <div class="survey-step" id="step-1-part1">

                            <p class="text-lg font-semibold mb-4">${data.title}</p>

                            ${data.options.map(option => `

                                <div class="flex items-center mb-2">

                                    <input type="${data.type}" id="${data.field}-${option.replace(/\s/g, '-')}" name="${data.field}" value="${option}" class="mr-2">

                                    <label for="${data.field}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                </div>

                            `).join('')}

                            ${data.nextText ? `

                                <p class="text-lg font-semibold mb-4 mt-6">${data.nextText}</p>

                                ${data.nextOptions.map(option => `

                                    <div class="flex items-center mb-2">

                                        <input type="checkbox" id="${data.nextField}-${option.replace(/\s/g, '-')}" name="${data.nextField}" value="${option}" class="mr-2">

                                        <label for="${data.nextField}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                    </div>

                                `).join('')}

                            ` : ''}

                            <div class="flex justify-between items-center mt-6">

                                ${backButton}

                                <button id="next-part1" class="btn-onward py-3 px-6 rounded-lg shadow-md" disabled>Next</button>

                            </div>

                        </div>

                    `;

                } else if (state.currentStep === 2) {

                    const data = surveyQuestions[state.role].part2;

                    html = `

                        <div class="survey-step" id="step-1-part2">

                            <p class="text-lg font-semibold mb-4">${data.title}</p>

                            <div class="flex flex-col space-y-4 mb-6">

                                ${data.options.map(option => `

                                    <div class="flex items-center">

                                        <input type="${data.type}" id="${data.field}-${option.replace(/\s/g, '-')}" name="${data.field}" value="${option}" class="mr-2">

                                        <label for="${data.field}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                    </div>

                                `).join('')}

                            </div>

                            <p class="text-lg font-semibold mb-4">${data.nextText}</p>

                            <div class="flex flex-col space-y-4 mb-6">

                                ${data.nextOptions.map(option => `

                                    <div class="flex items-center">

                                        <input type="checkbox" id="${data.nextField}-${option.replace(/\s/g, '-')}" name="${data.nextField}" value="${option}" class="mr-2">

                                        <label for="${data.nextField}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                    </div>

                                `).join('')}

                            </div>

                            <p class="text-lg font-semibold mb-4">${data.nextNextText}</p>

                            <div class="flex flex-col space-y-4 mb-6">

                                ${data.nextNextOptions.map(option => `

                                    <div class="flex items-center">

                                        <input type="checkbox" id="${data.nextNextField}-${option.replace(/\s/g, '-')}" name="${data.nextNextField}" value="${option}" class="mr-2">

                                        <label for="${data.nextNextField}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                    </div>

                                `).join('')}

                            </div>

                            <div class="flex justify-between items-center mt-6">

                                ${backButton}

                                <button id="next-part2" class="btn-onward py-3 px-6 rounded-lg shadow-md" disabled>Next</button>

                            </div>

                        </div>

                    `;

                } else if (state.currentStep === 3) {

                    const data = surveyQuestions.payment;

                    html = `

                        <div class="survey-step" id="step-1-payment">

                            <p class="text-lg font-semibold mb-4">${data.title}</p>

                            <div class="flex flex-col space-y-4 mb-6">

                                ${data.options.map(option => `

                                    <div class="flex items-center">

                                        <input type="${data.type}" id="${data.field}-${option.replace(/\s/g, '-')}" name="${data.field}" value="${option}" class="mr-2">

                                        <label for="${data.field}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                    </div>

                                `).join('')}

                            </div>

                            <div id="payment-details-container" class="hidden">

                                <p class="text-lg font-semibold mb-4">${data.nextText}</p>

                                <div class="flex flex-col space-y-4 mb-6">

                                    ${data.nextOptions.map(option => `

                                        <div class="flex items-center">

                                            <input type="radio" id="${data.nextField}-${option.replace(/\s/g, '-')}" name="${data.nextField}" value="${option}" class="mr-2">

                                            <label for="${data.nextField}-${option.replace(/\s/g, '-')}" class="text-sm">${option}</label>

                                        </div>

                                    `).join('')}

                                </div>

                            </div>

                            <div id="integrated-note" class="hidden text-sm text-gray-500 mt-2 p-3 rounded-lg bg-gray-100">

                                This option uses an in-app, commission-based payment model.

                            </div>

                            <div class="flex justify-between items-center mt-4">

                                ${backButton}

                                <button id="submit-survey" class="btn-onward py-2 px-6 rounded-lg shadow-md" disabled>Submit</button>

                            </div>

                        </div>

                    `;

                }

                appContainer.innerHTML = html;

                // Attach event listeners for the dynamically rendered content

                if (state.currentStep === 0) {

                    document.querySelectorAll('[data-role]').forEach(button => {

                        button.addEventListener('click', (event) => {

                            state.role = event.target.dataset.role;

                            state.currentStep = 1;

                            renderStep();

                        });

                    });

                } else if (state.currentStep > 0 && state.currentStep < 3) {

                    document.getElementById('back-btn')?.addEventListener('click', goBack);

                    document.querySelectorAll('input').forEach(input => {

                        input.addEventListener('change', validateAndEnableNextButton);

                    });

                    document.getElementById(`next-part${state.currentStep}`).addEventListener('click', () => {

                        saveStepResponses();

                        state.currentStep++;

                        renderStep();

                    });

                } else if (state.currentStep === 3) {

                    document.getElementById('back-btn')?.addEventListener('click', goBack);

                    document.querySelectorAll('input[name="paymentModel"]').forEach(input => {

                        input.addEventListener('change', (event) => {

                            const paymentDetails = document.getElementById('payment-details-container');

                            const integratedNote = document.getElementById('integrated-note');

                            if (event.target.value === 'Integrated payment processing (in-app payments)') {

                                paymentDetails.classList.add('hidden');

                                integratedNote.classList.remove('hidden');

                            } else {

                                paymentDetails.classList.remove('hidden');

                                integratedNote.classList.add('hidden');

                            }

                            validateAndEnableNextButton();

                        });

                    });

                    document.querySelectorAll('input').forEach(input => {

                         input.addEventListener('change', validateAndEnableNextButton);

                    });

                    document.getElementById('submit-survey').addEventListener('click', () => {

                        saveStepResponses();

                        saveSurveyResponse();

                    });

                }

            }

            function validateAndEnableNextButton() {

                const stepElement = appContainer.querySelector('.survey-step');

                const nextButton = stepElement.querySelector('button[id^="next-"]');

                const submitButton = document.getElementById('submit-survey');

                const inputs = Array.from(stepElement.querySelectorAll('input'));

                const isValid = inputs.some(input => input.checked);

                if (nextButton) {

                    nextButton.disabled = !isValid;

                }

                if (submitButton) {

                    submitButton.disabled = !isValid;

                }

            }

            function saveStepResponses() {

                const inputs = appContainer.querySelectorAll('input');

                inputs.forEach(input => {

                    const field = input.name;

                    if (input.type === 'checkbox') {

                        if (!surveyState.responses[field]) {

                            surveyState.responses[field] = [];

                        }

                        if (input.checked) {

                            surveyState.responses[field].push(input.value);

                        }

                    } else if (input.type === 'radio' && input.checked) {

                        surveyState.responses[field] = input.value;

                    }

                });

            }

            async function saveSurveyResponse() {

                if (!userId) {

                    console.error("User not authenticated. Cannot save responses.");

                    return;

                }

                const responseData = {

                    ...surveyState.responses,

                    timestamp: new Date()

                };

                // Save to private collection for user

                const privateResponsesPath = `artifacts/${appId}/users/${userId}/surveyResponses`;

                await addDoc(collection(db, privateResponsesPath), responseData);

                // Save to public collection for admin dashboard

                const publicResponsesPath = `artifacts/${appId}/public/data/surveySubmissions`;

                await addDoc(collection(db, publicResponsesPath), responseData);

                console.log("Survey responses saved successfully!");

                showView('confirmation');

            }

            // Event listeners for view switching

            adminLink.addEventListener('click', () => {

                showView('admin-login-form');

            });

            backToSurveyBtn.addEventListener('click', () => {

                renderStep();

            });

            startNewSurveyBtn.addEventListener('click', () => {

                surveyState.currentStep = 0;

                surveyState.responses = {}; // Clear responses

                renderStep();

            });

            document.getElementById('login-form-fields').addEventListener('submit', handleAdminLogin);

            logoutBtn.addEventListener('click', () => {

                showView('admin-login-form');

            });

            authenticate();

            startSlideshow();

            renderStep();

        });

    </script>

</body>

</html>